Purpose of a Core Model

Jeffrey Wix[1]

Introduction

At the ISO TC184/SC4 meeting in Berlin in 1993, the Building Construction group within WG3/T12 AEC commenced an Application Protocol Planning Project (APPP) with the objective of defining a framework for the development of Application Protocols (APs) suited to their needs and to determine the priority areas for initial AP development. Reporting to the Davos meeting in 1984, the APPP identified the need for a core capability which would enable sharing of common data between the various disciplines operating in building construction. At the Greenville meeting in 1984, New Work Item proposals were made for a number of APs and for a Building Construction Core Model (BCCM). Development of the BCCM and two APs were authorised at the Sydney meeting in 1995.

In developing the New Work Item proposal, there was some difficulty in determining where the BCCM should be placed within the STEP architecture since it proposed not only an Application Resource but also an exchange capability. The decision was taken that it should be considered as an Application Resource and was designated as part 106.

At the same time as the BCCM idea was germinating, other groups within STEP were thinking about a similar concept. The idea of an overall AEC Core Model was discussed in some detail leading up to a meeting in London in February 1995. A number of the ideas discussed at this meeting have affected the development of the BCCM.

More recently, the concept of a Core Model for Engineering Analysis has been discussed.

Even without the above, the notion of a Core Model may be seen to be implicit in some of the work already being undertaken within STEP. For instance, AP214 developed for the automobile industry uses the words 'Core Data' in its title. The development of 'building blocks' by the ESPRIT Maritime project is influencing the development of APs within the shipbuilding sector. The set of building blocks available may potentially considered as a type of Core Model.

Recent discussion on BCCM development has raised a number of issues which reflect the need for determination of the positioning of Core Models within the overall technical architecture of STEP. Analysis of these issues suggests that there may be a gap in the current technical architecture of STEP which could be addressed by the current work of the WG10 Architecture and Methodology group and, in so doing, fix the purpose of Core Models and provide a basis for their future development.

A preliminary version of this paper was presented at the ISO TC184/SC4 meeting in Dallas in January 1996. This paper is a final version prepared at the request of the WG10 chair.

[1] Project Leader for the proposed ISO 10303 part 106 - Building Construction Core Model.

Background

Within the current architecture of STEP, data models are developed either as Integrated Resources or as Application Protocols. The content of each of these is defined (1) as follows:-

> Integrated Resources (IRs) define the data content which provides the basis for Application Protocols. These models include those which have general applicability ('40' series) and those which support a particular application or class of application (100 series). The product data in a Resource Information Model is encapsulated in an implementation independent form and is only implemented indirectly via an Application Protocol.
> Application Protocols (APs) provide a specific functionality for an application requirement such that the data content of each is self contained and complete. APs state the information needs of a particular application, specify an unambiguous means by which information is to be exchanged for that application and provides conformance requirements and test purposes for conformance testing. APs are within the 200 series of standards within ISO 10303.

An AP is developed initially as an Application Reference Model (ARM) which details the application requirement in terms of the entities, relationships and business rules which are within the stated scope. Development of the ARM is typically undertaken by persons who have a knowledge of the application requirement, usually in conjunction with persons having a knowledge of data modelling methods, and takes the form of a graphical model annotated according to terminology in normal use within the application or an acceptable abstraction of such terminology.

On completion of the ARM, and after its acceptance by a representative community of its potential users, the model undergoes interpretation. This is a process whereby the model is transformed so as to use entities defined within the integrated resources wherever possible. The resulting Application Interpreted Model (AIM) differs significantly from the ARM although a mapping table is developed which identifies how elements of the ARM are transformed to the AIM.

There does not appear to be a mechanism which allows for a reverse transformation which would enable the derivation of an ARM from the AIM other than by reverse engineering the mapping table. Thus, within the present architecture of STEP, interpretation appears to be an irreversible process. Since the standard is defined according to the AIM and since exchange should occur according to the AIM, the ARM being only for information within the final standard, irreversibility should not be an issue.

Interpretation then provides for an exchangeable form of the AP and makes available to it the general capability of the IRs.

Application Requirement

Application requirements within STEP are identified normally according to a precise industrial context or domain and adopt a scope which is the same as or falls within that context (2) . Thus, within a structural engineering context, an AP may have a scope of structural steel frameworks which is within that context. Alternatively, within an HVAC context, an HVAC AP may have a scope which is equal to the context. Strictly, the contexts of these examples do not overlap and consequently the scopes also do not overlap.

Because of the contextual nature of AP development, there are no mechanisms existing within the current architecture which enable the exchange of information between Application Protocols. The only way in which this could be done is to develop an AP whose context and scope overlaps domains. At this time, no such developments are known other than those proposed within AP221.

Common Interpretation

Whilst context and scope do not overlap, there are elements of different models which are intended to fulfil the same or similar functions. These 'shared requirements' are currently dealt with in STEP by Application Interpreted Constructs (AICs).

The shared requirement demonstrated by an AIC may be considered as a point of overlap between APs. However, their provision is aimed at delivering a single interpretation of a model segment which is common to more than one AP and does not provide for the exchange of information between those APs.

According to current methodology, if APs have several shared requirements then each shared requirement would be interpreted as a separate AIC. Thus, if there is a lot of shared requirement between models, there could be a substantial number of AICs.

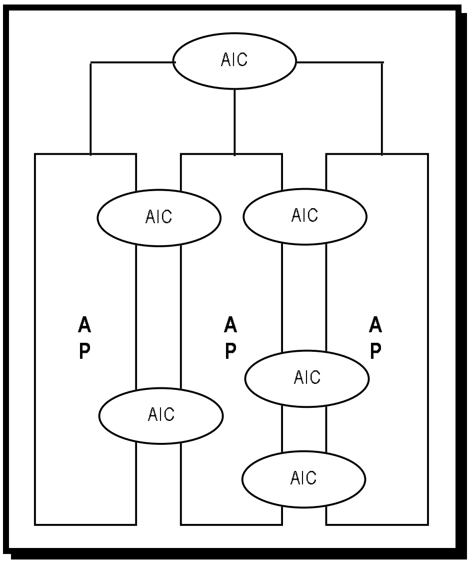


Fig.1 Common interpretation

Presently, AIC development tends to be undertaken after development of an ARM and occurs as a result of shared requirements being perceived. The AIC, as its name suggests, is an interpretation. That is, it is used as part of an AIM. There is no requirement for any common model construction within ARMs.

The Building Construction Application Protocol Planning Project

At the STEP meeting in Davos in May 1993, the Building Construction Group presented an Application Protocol Planning Project which set out a co-ordinated plan for AP development and identified those APs which should have development priority. It recognised that the building construction industry is made up of a number of disciplines each of which have their own application requirement.

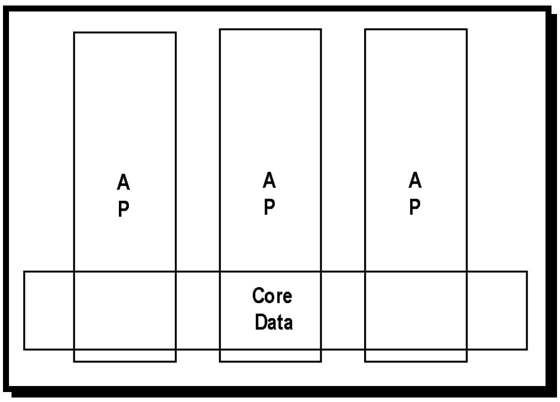


Fig.2 Core data

It also identified that there is a set of information which needs to be exchanged between disciplines. This set of information, described as being of common interest, is less deep for any given application than would be the case for an AP but would have the capability to describe the means of exchanging data between application requirements. Such data were referred to as Core data and it is from this that the concept of a Building Construction Core Model has developed.

Whilst drawing on the planning work carried out by the Building Construction Group, the above idea is not considered to be exclusive to building construction. Other industries which have multiple AP requirements and which operate in multi-disciplinary mode are likely to reflect similar requirements.

An AEC Core Model

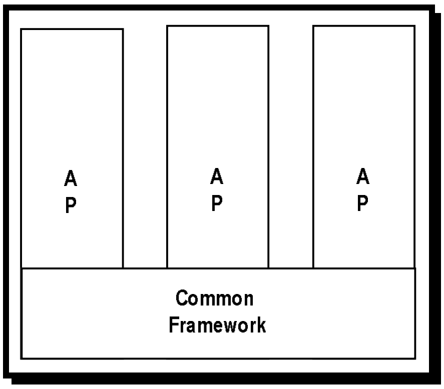


Fig.3 Common framework

At the same time as the idea of a Building Construction Core Model was developing, the concept of a more generic AEC Core Model was also being proposed. This drew on the work of the EPISTLE process industry group and derived its influences from various sources including work by Shell [(2), (3), (4), (5)] and Gielingh [(6)]. Differing from the building construction approach, the AEC Core Model has been proposed as a framework for the development of APs using a Generic Entity Framework

Core Model Concepts

The above demonstrates three ideas of Core Models:-

1. Interpretation of common requirement.
2. Specification of common data.
3. Development of a common framework.

Two methods of developing information models have been identified, namely top down and bottom up. These are also referred to [(9)] as axiomatic and heuristic. The ideas of Core Model development can be mapped to these methods:

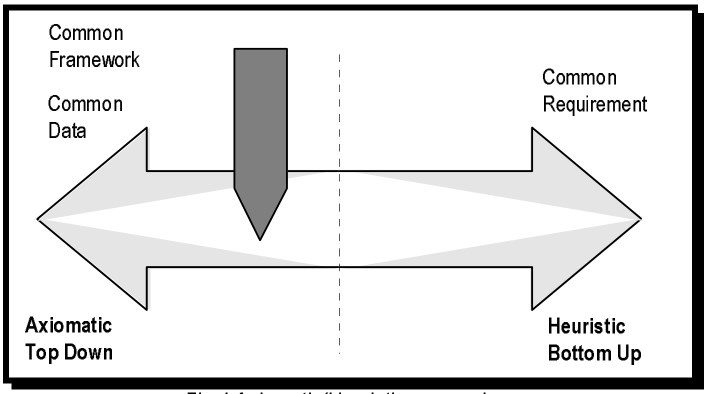


Fig.4 Axiomatic/Heuristic approaches

Common Requirement	Bottom up
Common Data	Top Down
Common Framework	Top Down

It is reasonable to ask two questions concerning these methods:-

1. Are they exclusive?
2. To what extent are they different (or the same)?

The purpose of a framework is to provide a consistent basis for the development of data models [3]. Without a consistent basis, models produced in isolation will vary arbitrarily and be difficult to reconcile. Given a consistent basis, independently developed models will tend to be consistent.. The Building Construction Core Model and the EPISTLE Core Model provides such a framework through a subtype/supertype hierarchy, although there are slight differences in approach.

The purpose of a common data specification is to identify the information which is of common interest between the various disciplines within an industry sector and consequently between the various APs which serve that sector. Such common data is intended to inform one discipline of the proposals of another discipline in a form which is relevant and comprehensible. For instance, information concerning a structural steel framework exchanged between one structural engineer and another can be complete and comprehensive. However, information concerning the same structural steel framework being exchanged between a structural engineer and other disciplines can be more limited, being composed of such characteristics as shape, size, location, support capability, material etc. The important aspect of common data is that it should detail only that which is relevant to described needs. It is a problem which is well understood (if not always well handled) in graphic data exchange using specifications such as IGES and DXF where the information given on a specialist drawing is far more than is required for a non specialist. To overcome this, methods have been adopted which allow partitioning of information so that it can be limited in data exchange. The most significant of these has been the development of layer conventions which allow for simple or complex arrangements of graphical information according to requirements.

Common data is described as primarily a top down approach (the vertical arrow in the diagram representing the approximate position in an axiomatic/heuristic scale [9]..The fact that it is positioned part way along the scale identifies that it may also be partially bottom up. This implies that a common requirement is determined within two or more APs and that this common requirement is promoted to common data.

Common requirements are determined in a similar manner to AICs. However, they are not AICs in a Core Model sense since this would imply that they are ARM developments. Thus, whilst a need for ARM level common requirements within a Core Model can be perceived, there is no specific provision for them within the current STEP architecture. For this, an Application Reference Construct (ARC) equivalent to an AIC might be appropriate and it is on this basis that common requirements are further considered. Note that designation of this concept as an ARC is for the purpose of this paper only and is used to maintain consistency of nomenclature.

The concept of an ARC can be seen not only at a Core Model level at which it can enable the exchange of limited data and further specialisation through its use as a template but also at more detailed levels where the 'brick' idea developed within the EU MARITIME project and used within the shipbuilding APs can be seen to correspond. Oehlmann [7], in correspondence with West refers to the AP concept and conformance classes as below:-

> 'The benefit of data sharing is the ability to maintain a single, consistent context during the product life cycle. Consequently, especially configuration management concepts and associations between different model domains (e.g. different systems) can be handled more effectively. The STEP methodology, targeted towards data exchange and the interaction between homogeneous model domains (the AP concept) seems at least not "ideal" in this respect. Further, STEP enforces the identification of conformance classes within the standard, assuming that such classes can be identified beforehand, that their number is limited, and that the borders of such classes do not cut between entity references. We believe that these assumptions do not reflect reality in a good way.'

Thus, we can determine that:-

- A common framework sets the basis for consistent modelling.
- Common data enable exchange of information between APs.
- Common requirements provide the capability for reuse of model constructs.

In consequence, the three approaches are not exclusive; in fact, in combination, they can offer a more powerful core capability and a better response to industrial requirements.

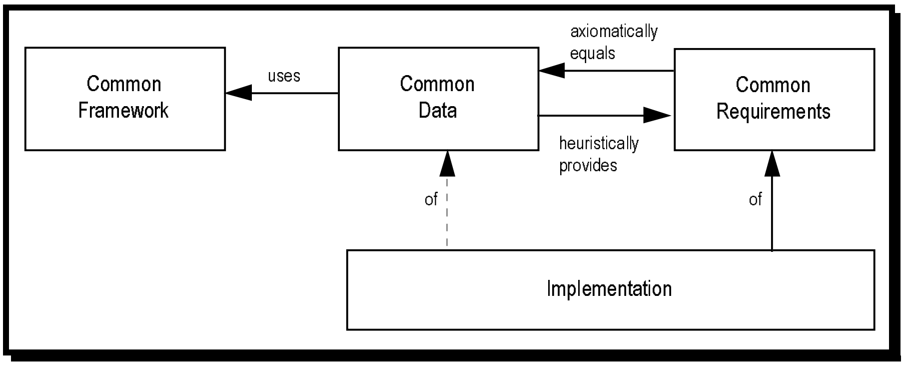


Fig.5 Combination of approaches

Further, we can identify that common data requires the provision of a common framework and may be developed from common requirements either prior to AP development (axiomatically) or post AP development (heuristically).

ARM Level Integration

Any model described in EXPRESS can be exchanged or shared. Practically, it does not matter if that model uses industry terminology as an ARM or interpreted terminology as an AIM. Since EXPRESS is intended to be an unambiguous specification, any applications which reflect a given model can exchange or share data. It is probably true to say that, currently, there are more ARM level implementations using STEP technology than there are AIM level implementations using STEP. Evidence of the potential for successful ARM level implementations is available from the CIMsteel project which defines an industry agreed model for the exchange of structural steelwork data.

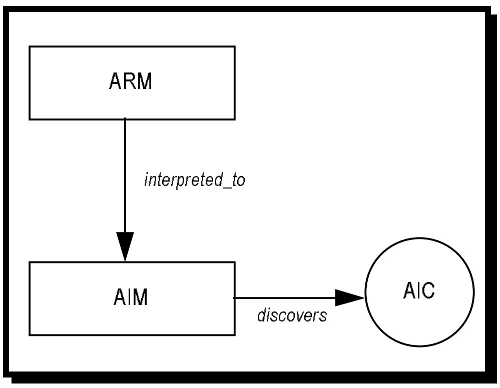


Fig.6 Current Architecture

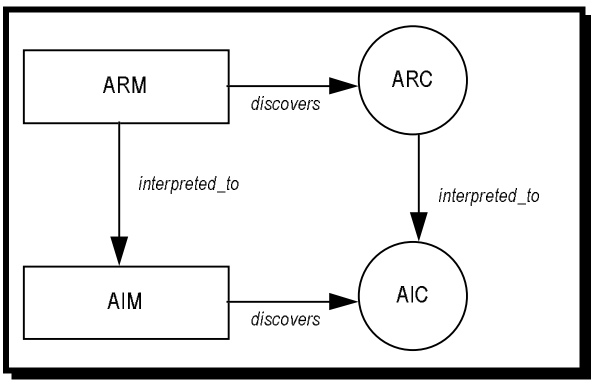


Fig.7 Proposed Architecture

If it is possible to have successful ARM level industry implementations, it should be equally possible to have ARM level industry integration. Yet the current methodology of STEP does not appear to facilitate this.

A significant advantage to the idea of an ARC is that it should be possible to interpret an AIC from it so that there is a correlation between the industry level and the interpreted level equivalent to the ARM/AIM correlation. This has the advantage of exposing the AIC to a level of industry scrutiny not currently available.

The ARC/AIC Conundrum

On the assumption then that an ARC and an AIC may be defined equivalently and that they can be used both to provide an exchange mechanism as well as identifying a shared requirement, the conundrum exists as to whether an ARC is developed from an AIC as a result of identifying shared requirements at interpretation or whether an AIC can be directly interpreted from a prior definition of an ARC.

In truth, the answer should be both. Given the range of industrial expertise which ought to be available within STEP, it is inconceivable that many common requirements and consequently common data cannot be identified for an industry domain axiomatically. It defies belief that industry practitioners who communicate with each other with some level of success in the absence of STEP cannot figure out how they achieve it for the definition of the very standard which proposes to amplify their communication capability.

However, it may not be possible to state the complete range of common requirements and common data at a first attempt (i.e. in a totally top down manner) for various reasons including:-

- Not all of the complete set of APs required for an industry domain will be under development at one time. Within building construction, initial efforts are for HVAC and structural steel frameworks; spaces are proposed to follow along with project management; reinforced concrete may come later, detailed architectural capabilities are not even under discussion at present, neither are power and lighting, control systems, waste and drainage, acoustics any form of civil engineering and etc. Later APs may well indicate additional common requirements not initially identified which will need to be included heuristically in a Core Model as a later version.
- Common requirements at higher levels of abstraction may not be immediately apparent in the axiomatic identification of common data.

Common Data versus ARC/AIC

A Core Model can have at least two different forms, as a model to support the Core business or as a kernel model for integration and a basis for specialisation's in AEC areas and object (entity) types. From an offshore industry point of view, the second definition is the major issue [(8)].

Within building construction, the kernel is required for exactly the same reasons as the offshore industry and this is seen as equivalence to the integrated application resources.

The model to support Core business ultimately depends on how that Core business is defined. It might be defined as the envelope surrounding all the data which is fundamentally necessary to the carrying out of a business. In this case, the Core Model would tend to be massive and act as a unitary mechanism for a whole industrial need. This might well be appropriate for an industry which operates with close control.

Alternatively, it might be defined in the context of STEP as supporting the core exchange and sharing requirements of that industry in which case we would be concerned with a fragmented industry under diverse management. This approach is seen as more relevant to building construction since it allows sharing of relevant information between disciplines without the need for the provision of unnecessarily detailed, extraneous or redundant information.

Both of the Core business scenarios indicate the requirement for an implementable exchange capability in addition to the provision of a Common Requirements facility..

Current and Future Interpretation

Assume now that a Core Model exists in a form which has elements of kernel, common requirements and common data. Assume also that it has been interpreted so that it is fully a STEP part. An argument for a Core Model is that, having been interpreted, use of its facilities by APs provides pre-interpretation possibilities and, in so doing, makes the AP development process more efficient. However, some questions remain concerning AP development and integration.

1. Is an AP simply a collection of parts from a Core Model?

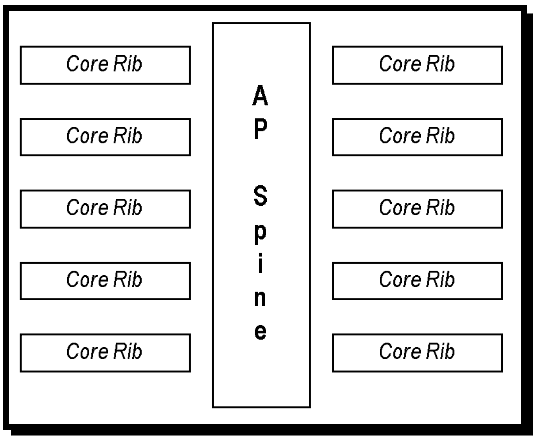
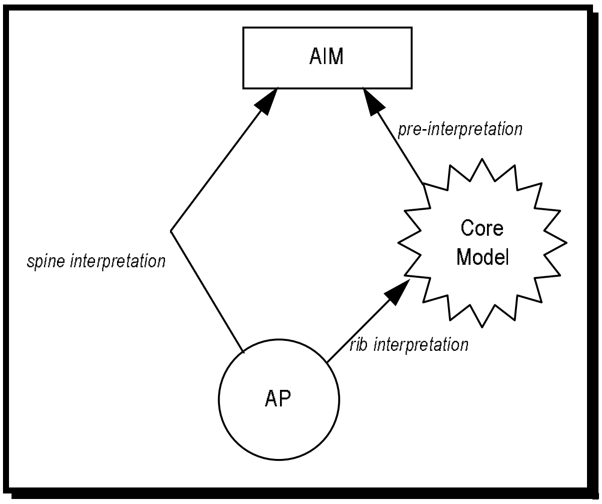


Fig.8 Spine and Rib AP

Whilst an AP may draw significantly upon the resources of a Core Model, it is unlikely to do so in entirety. If it did so, then the point must be raised as to 'why develop the AP at all since everything required is already available'. Clearly, every AP has a spine of information which is absolutely specific. It can then be supported by 'rib' resources.

2. Must an AP use pre-interpretation?



If pre-interpretation is a possibility and its use is efficient, then it ought to be used. However, the spine of the AP may require specific interpretation according to the current architecture (since the facilities would not be in a Core Model) and this should remain a possibility.

Fig.9 Interpretation routes

Conformance Testing at the ARM Level

"STEP should be a data driven Standard that doesn't require any more data modelling to implement. In the AEC world, all the useful implementations to date have been at the ARM level. Therefore, it would seem sensible to define an appropriate mechanism for conformance testing at the ARM level" (10).

So What is a Core Model?

The above has identified a number of things that a Core Model should be able to do.

1. It should be able to provide a kernel equivalent to current application resources
2. It should be able to define common requirements equivalent to an AIC at a Reference level (ARC)
3. It should be able to describe data to be shared between disciplines (i.e. between APs)
4. It should be able to support pre-interpretation of an AP without restricting the ability to do post interpretation
5. It should be able to develop integration at the ARM level in a standardised way as a precursor to complete STEP interpretation.
6. It should be able to be conformance tested at the ARM level
7. It should be able to enable fast implementation development and deliver a response to external challenge.

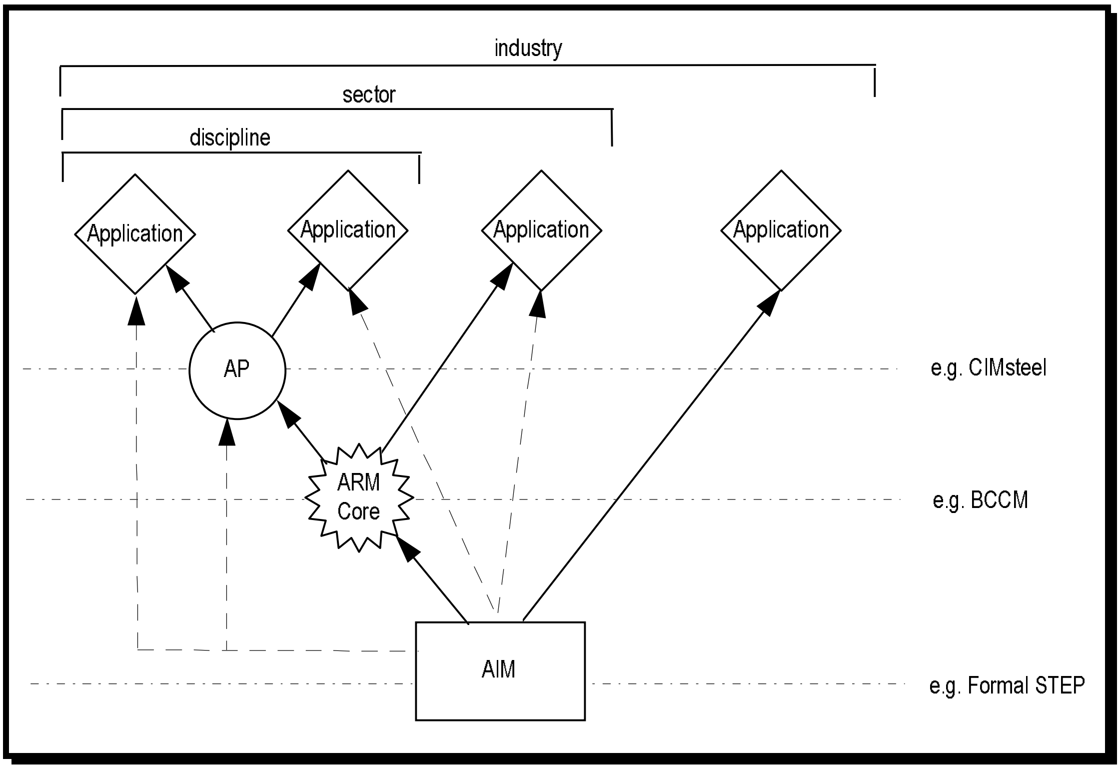


Fig.10 Core model purpose

This does not say that a Core Model must be able to do all of these things. Undoubtedly, further thought needs to be given. However, this paper presents the view that Core Models should form an integral part of the overall STEP architecture and that they ask questions of STEP which need to be answered.

References

1. ISO 10303-1, Industrial Automation Systems - Exchange of Product Model Data - Part 1: Overview and Fundamental Principles, ISO, Geneva, 1995
2. West M., Developing High Quality Data Models. Volume 1: Principles and Techniques, Shell International Petroleum Company Ltd., London, March 1994
3. West M., Developing High Quality Data Models. Volume 2: The Generic Entity Framework. , Shell International Petroleum Company Ltd., London, March 1994
4. West M., Developing High Quality Data Models. Volume 3: Data Model Templates, Shell International Petroleum Company Ltd., London, March 1994
5. Ottman B., West M and Fyfe S., Reviewing and Improving Data Models, Shell Internationale Petroleum Maatschappij B.V., The Hague, December 1992
6. Gielingh W., General AEC Reference Model,
7. Oehlmann R.., MARITIME Requirements on Model Contents, Development and Usage, Private Communication to M.West, 15th May 1995
8. Myklebust O.,Offshore Assessment of Core Model Activities, AEC Core Model Workshop, London, 6 February 1995
9. West M., Presentation on Development of an AEC Core Model, AEC Core Model Workshop, London, 6 February 1995
10. Lord S., Private Communication, January 1996

Acknowledgements

The following are acknowledged for their contributions to the development of this paper:-

Frits Tolman and Robert Los, TNO (NL), editors of the BCCM, Kjell Svensson, KTH (Sw), Robert Amor, BRE (UK), Thomas Liebisch, CAB (G), Wolfgang Haas, (Haas & Partner (G), Richard Wiitenoom, AusSTEP (Aus), Howard Leslie, CSIRO (Aus), Robin Drogemuller, James Cook Univ (Aus), Stuart Lord, ICI (UK), Grahame Cooper, Salford Univ (UK), Mike Ward, Univ. Of Leeds (UK), Julian Fowler, PDT (UK).

This paper has been prepared within the Computerised Exchange of Information in Construction project, a project supported by the UK Department of the Environment under its Partners in Technology program.

For further information, contact David Bloomfield at BRE on ‘bloomfieldd@bre.co.uk’ or Jeffrey Wix at ‘100342.2537@compuserve.com’.